UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1278

*Programme 1*
*Programmation, Calcul Symbolique*
*et Intelligence Artificielle*

# ON THE GENERIC RIGIDITY OF PLANE FRAMEWORKS

**Henry CRAPO**

**Août 1990**

*RR.1278*

# Sur la rigidité générique des réseaux plans

HENRY CRAPO

*Bât 9, INRIA, B.P. 105, 78153 Le Chesnay, Cédex, France*

première distribution 14/10/88,

revisé, 14/8/1990

*Resumé.* Un nouveau type d'analyse des réseaux plans à barres et à joints fournit une caractérisation élémentaire des structures 2-isostatiques en position générale. Cette caractérisation est rendue effective par une adaptation de l'algorithme de Edmonds (pour le recouvrement d'un matroïde par des ensembles indépendants disjoints).

---

# On the Generic Rigidity of Plane Frameworks

HENRY CRAPO

*Bât 9, INRIA, B.P. 105, 78153 Le Chesnay, Cédex, France*

first distributed 14/10/88,

revised, 14/8/1990

*Abstract.* A new type of analysis of plane bar and joint frameworks provides a simple characterization of generically 2-isostatic structures. This characterization is rendered effective by an adaptation of Edmonds' algorithm for coverings of a matroid by disjoint independent sets.

# On the Generic Rigidity of Plane Frameworks

*by Henry Crapo*

*Bât 9, INRIA, B.P. 105, 78153 Le Chesnay, Cédex, France*

first distributed 14/10/88,

revised, 14/8/1990*

*Abstract.* A new type of analysis of plane bar and joint frameworks provides a simple characterization of generically 2-isostatic structures. This characterization is rendered effective by an adaptation of Edmonds' algorithm for coverings of a matroid by disjoint independent sets.

## 1. Introduction

In this article we present a simplified characterization of graphs of bar and joint frameworks which are generically 2-isostatic (minimal statically rigid in the plane):

**Theorem 1.** *A graph $G = (V, E)$ is generically 2-isostatic if and only if the set $E$ of edges of $G$ is the disjoint union of three trees $T_i$ such that each vertex of $G$ is incident with exactly two of the trees $T_i$, and with a further property that subtrees of distinct trees $T_i$ do not have the same span.*

This characterization permits us, after a single application of a slightly modified form of Edmonds' algorithm (for minimal partitions of a matroid into independent sets), to decide whether a given graph $G$ is generically 2-isostatic. (See Section 5, Algorithm 11. The proof of Theorem 1 will be given below, in section 4.)

* The author's research is supported by the European Community, Esprit Basic Research Action Number 3075 (ALCOM) and by the Natural Sciences and Engineering Research Council, Canada.

An *implicit edge* in a generic framework with graph $G$ is a pair $ab$ of vertices of $G$ which will support an external equilibrium load applied along the line $ab$. In Section 8, a further adaptation of the algorithm decides (on purely combinatorial grounds) which pairs $ab$ of vertices of a graph are implicit edges in a generic plane framework. This leads to an algorithmic construction of Henneberg 2-sequences for the construction of any given generically 2-isostatic graph.

In Section 9, we consider possible extensions of the algorithm to bar and joint frameworks in 3-dimensional space.

## 2. Definitions

A *graph* $G = (V, E)$ is a finite set $V$, whose elements are called *vertices*, together with a set $E$ of (unordered) pairs of distinct vertices. These ordered pairs $a, b$ are called *edges*, and the vertices $a, b$ in such a pair are called the *ends* of the edge. An edge $a, b$ is denoted more simply as $ab$. A *subgraph* $G'$ of a graph $G = (V, E)$ is a pair of subsets $(V', E')$ where $V'$ is a subset of $V$, $E'$ is a subset of $E$, and every end of every edge in $E'$ is a vertex in the subset $V'$. (It follows that a subgraph of a graph is a graph in its own right.) For any subset $V'$ of the vertex set $V$ of a graph, the *induced subgraph* $G|_{V'}$ is the graph $G' = (V', E')$, where $E'$ is the set of edges both ends of which are in the subset $V'$. A *path* in a graph is a subgraph consisting of a set of distinct vertices $a, b, \ldots, e, f$ and edges $ab, bc, \ldots, ef$. We say that such a path is a path *from $a$ to $f$*. A graph $G$ is *connected* if and only if, for every pair $\{a, f\}$ of distinct vertices, there exists a path in $G$ from $a$ to $f$. A *cycle* in a graph is a subgraph consisting of a set of at least three distinct vertices $\{a, b, \ldots, e, f\}$ and edges $ab, bc, \ldots, ef, fa$. A set of edges is *independent* if and only if it contains no cycle. It is a *tree* if and only if it is a connected graph which contains no cycle. Note that a single vertex of $V$, with no edges, constitutes a tree in a graph $G = (V, E)$.

The *span* $\langle A \rangle$ of a set $A$ of edges of a graph is the set $A$ together with all edges $x \notin A$ for which there exists a cycle $C$ containing $x$ in the set $A \cup x$. In particular, a *spanning tree* for a graph $G = (V, E)$ is a tree whose span is the entire set $E$. The *rank* $r(A)$ of a set $A$ of edges is the number of edges in any independent subset $A'$ which spans $\langle A \rangle$. Thus, for a graph $G$ with $v$ vertices and c connected components, $r(G)$ is the number of edges in an independent set which spans $G$, that is, $r(G) = v - c$. If a set $A$ of edges is equal to its own span, it is *closed.*

A graph G = (V, E) is *generically 2-rigid* if and only if it has at least one realization as a bar and joint framework in the plane wherein it resolves all static equilibrium loads (equivalently, a realization in which it has no non-trivial infinitesimal motion). If such a graph has $v$ vertices, it must have at least $2v - 3$ edges. Those generically 2-rigid graphs with exactly this minimum number of edges are called *generically 2-isostatic.* We refer the reader to the Tay-Whiteley article [$T_1$] for definitions of "equilibrium loads", "infinitesimal motions", and such terms. For the purposes of this paper, we will rely on Laman's theorem (Theorem 2, below) as a characterization of generically 2-isostatic graphs, and we will have no occasion to deal directly with static or mechanical properties of frameworks.

## 3. Background

Generically 2-isostatic graphs were first characterized by Laman [$L_1$], and Asimov and Roth [A], in a form analogous to the statement of Hall's matching theorem.

**Theorem 2. (Laman's Theorem)** *A bar and joint framework is generically isostatic in the plane if and only if*

*(1) it consists of exactly $2v - 3$ bars on $v$ vertices, and*

*(2) for every subset of $v'$ vertices (for $2 \leq v' < v$) the induced subgraph has no more than $2v' - 3$ bars.* □

Statement (1) of Laman's Theorem asserts that a generically isostatic plane framework has just enough bars to reduce the $2v$ degrees of freedom of $v$ free vertices in the plane to the 3 degrees of freedom of a single rigid body. Statement (2) says there are not too many bars concentrated on any proper sub-framework. If too many bars *were* so concentrated, the rest of the framework would necessarily be underbraced.

In somewhat similar combinatorial problems, simpler characterizations have been found in terms of "cospanning trees". These characterizations are better suited to algorithmic verification, thanks to Edmonds' algorithm [E]. Lovasz and Yemeni [$L_2$] produced such a characterization of generically 2-isostatic graphs: a bar and joint framework with graph $G = (V, E)$ is generically isostatic in the plane if and only if, for every pair $(a, b)$ of vertices of $G$, the enlarged graph $G_{a,b}$ with an edge at $ab$ (or a doubled edge, if $ab$ was already an edge in $G$), is the edge-disjoint union of two spanning trees. In principle, the search for cospanning trees is more efficient than a test of cardinality pursued with respect to all subsets of the set of vertices. Unfortunately, this potential gain in efficiency is vitiated by the necessity to test for cospanning trees with respect to a rather large set of augmented graphs. Recski [$R_1$] reduced the necessary search to those graphs $G_{a,b}$ such that $ab$ is an edge of $G$. Subsequently, Imai [I] gave an algorithm which makes only a single search for a maximal matching on a bipartitite graph associated with the graph $G$. That all generically 2-isostatic graphs (and only those graphs) can be constructed recursively by Henneberg 2-sequences, was proven by Tay and Whiteley [$T_1$], in a paper which also surveys the previous literature on the subject, and studies possible generalizations of the Henneberg methods to generically 3-isostatic graphs. One of the referees of this article kindly pointed us to related work by Gabow and Westermann [G&W]. We discuss the connections with their work in Section 10.

This present work is the result of a fortunate misunderstanding between the author and Stéphane Roux, a physicist using percolation theory to study the critical states of various materials [G]. Roux understood me to say that a graph is generically 2-isostatic if and only if it is the edge-disjoint union of three trees, exactly two of which meet at every vertex. He planned to apply analogous reasoning to problems of propogation in antiferromagnetic materials [$R_2$]. It was his correct intuition that the "three trees" characterization is at least "almost correct" (together with the difficulty of finding a counterexample) that encouraged me to undertake the present research.

## 4. Concerning $3T2$-graphs

**Definition 3.** *An $nTk$-graph $(G, T_i)$ is a graph $G$ whose edge set is expressed as the disjoint union of $n$ trees $T_i$ , such that every vertex of $G$ is in precisely $k$ of those trees.*

**Proposition 4.** *Any $nTk$-graph with $v$ vertices has $kv - n$ edges.*

Proof: Assume the trees $T_i$ in an $nTk$-graph $(G, T_i)$ have $v_i$ vertices, and thus $v_i - 1$ edges, respectively. Since $\sum v_i = kv$, the graph has $\sum(v_i - 1) = kv - n$ edges. □

We will be dealing primarily with $3T2$-graphs. Figure 1 provides a few typical examples. Tree $T_1$ is shown with wide black edges, $T_2$ with gray edges, and $T_3$ with thin black edges. The simplest such graph, (D), consists of a single edge; in this case two of the three trees consist of single vertices.

For graphs with $2v - 3$ edges on $v$ vertices, the condition "$3T2$" can be stated in an apparently weaker, but equivalent form, as follows.

**Proposition 5.** *Assume that a graph $G = (V, E)$ with $v$ vertices has $2v - 3$ edges, and that $C_i$, for $i = 1, 2, 3$, are connected subgraphs of $G$ which partition the edge set $E$ of $G$, such that every vertex of $G$ is in at least two of the subgraphs $C_i$. Then $(G, C_i)$ is a $3T2$-graph.*

Proof: Assume the subgraphs $C_i$ have $v_i$ vertices, respectively. Since they are connected, they have $e_i \geq v_i - 1$ edges. Since the $2v - 3$ edges of $G$ are partitioned into these three sets,

$$2v - 3 = \sum e_i \geq \sum(v_i - 1),$$

and $2v \geq \sum v_i$. The average (over $V$) of the number of incident subgraphs $C_i$ is less than or equal to 2. Since every vertex is incident to at least two of the subgraphs $C_i$, every vertex is incident with exactly two of the $C_i$. So $2v = \sum v_i$, and $\sum e_i = \sum v_i - 1$. Thus $e_i = v_i - 1$ for each index $i$, and all the connected subgraphs $C_i$ are trees. □

**Definition 6.** *A $3T2$-graph $(G, T_i)$ is proper if and only if there is no pair $J, K$*

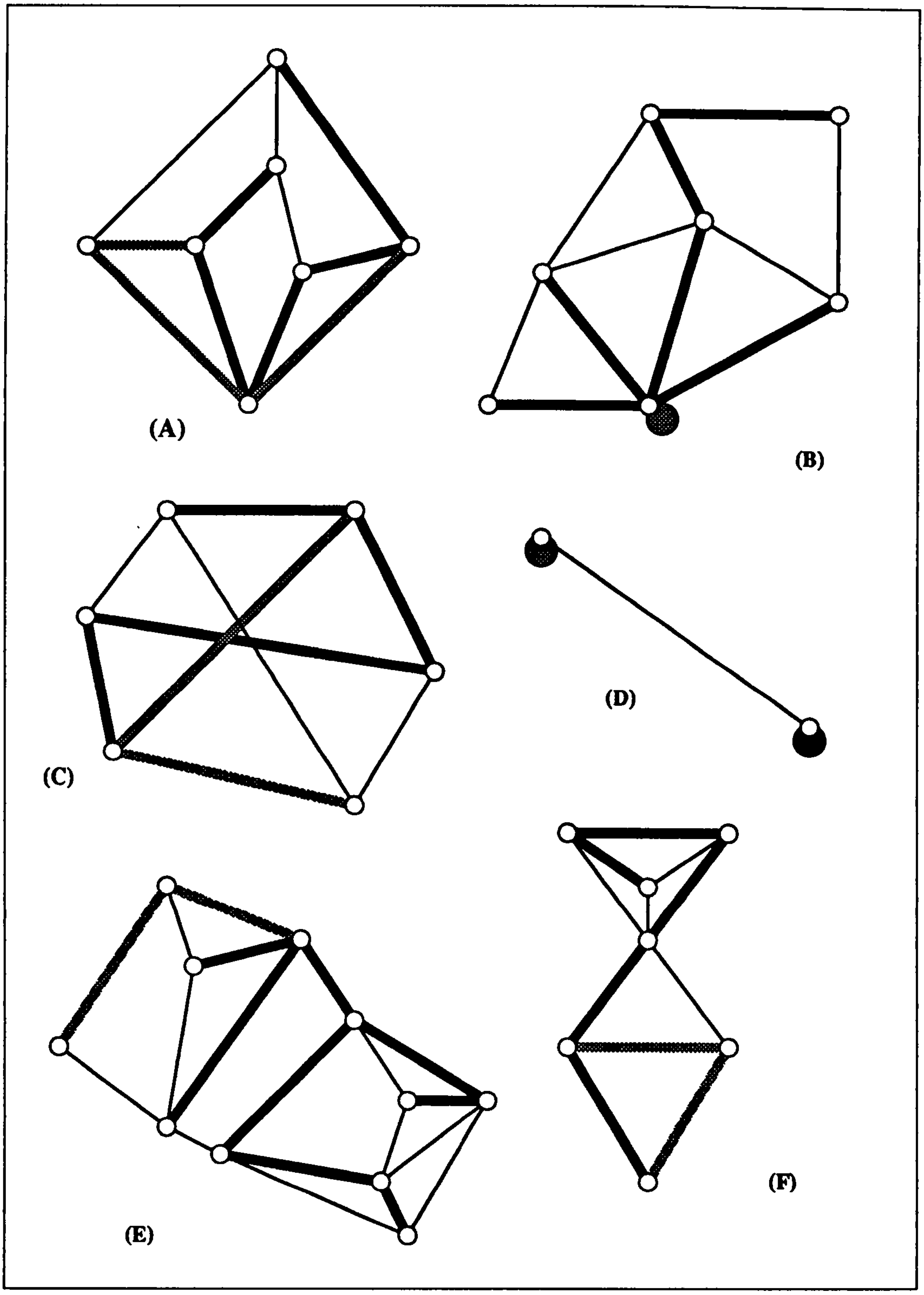


Figure 1. 3T2 structures on various graphs

*of trees, subtrees of distinct trees $T_i$, each having at least one edge, and having the same span, $\langle J \rangle = \langle K \rangle$.*

The $3T2$-graphs (E) and (F) are not proper: they have subgraphs on 6 and 4 vertices, respectively, which are spanned by distinct subtrees of $T_1$ and $T_3$. (It is also clear that these two graphs are not generically 2-rigid.)

**Restatement of Theorem 1.** *A $3T2$-graph $(G, T_i)$ is generically 2-isostatic if and only if it is proper.*

Proof: If the graph $G$ is not generically isostatic in the plane, then, by Laman's theorem, there is an induced subgraph $G' = (V', E')$ with $v'$ vertices, $v' \geq 2$, and at least $2v' - 2$ edges. The corresponding induced subgraphs $T_i'$ are independent sets with $v_i'$ vertices, $c_i'$ components, and thus $v_i' - c_i'$ edges, respectively. Every vertex in $V'$ is incident with an edge in exactly two of these independent sets, so $G'$ has $\sum(v_i' - c_i') = 2v' - \sum c_i'$ edges. At least two of the integers $c_i$ are positive. Since $G'$ has at least $2v' - 2$ edges, the sum $\sum c_i'$ is at most equal to 2. Thus two of the independent sets $T_i'$ are trees, and the third has no vertices. $G'$ is cospanned by two of the three trees $T_i$.

Conversely, if $G'$ is an induced subgraph of $G$ spanned by subtrees of two of the three trees $T_i$, then it has at least $2v' - 2$ edges, and the graph $G$ does not satisfy the conditions of Laman's theorem. It is not generically 2-isostatic. □

By Theorem 1, the test as to whether a given graph $G$ is generically 2-isostatic can be carried out in two steps. First, decide whether $G$ is $3T2$. If no, it is not generically 2-isostatic. If yes, output such a representation $(G, T_i)$. Second, decide whether $(G, T_i)$ is proper. $G$ is generically 2-isostatic if and only if *this* $3T2$-structure is proper. We shall see that both of these tests can be carried out in a single application of Edmonds' algorithm.

A minor modification (replacing "trees" by "connected subgraphs") of Theorem 1 provides a characterization of generically 2-rigid frameworks (rigid, but not necessarily independent).

**Theorem 7.** *A graph $G$ is generically 2-rigid if and only if it is the edge-disjoint union of three connected subgraphs $G_i$ of $G$ such that*

*(1) every vertex of $G$ is in exactly two of those subgraphs*

*(2) no subgraph of $G$ having at least one edge is spanned by subgraphs of two of the three subgraphs $G_i$.*

Proof: Assume that a graph $G$ together with three connected subgraphs $G_i$ satisfy the two conditions given in the statement of the theorem. Successive deletion from $G_i$ of edges which, at each stage, are contained in a cycle in $G_i$ will reduce $G_i$ to a tree $T_i$. Do this separately for $i = 1, 2, 3$. The three trees $T_i$ have the same incidence with vertices of $G$ as has the original subgraphs $G_i$, so they represent $G' = \cup T_i$ as a $3T2$-graph. Subgraphs of $G$ spanned by subtrees of two of these trees $T_i$ are also spanned by those subtrees as subsets of the graphs $G_i$. Since this does not happen, $G'$ is a proper $3T2$-graph, and is generically 2-isostatic. Since $G$ contains $G'$ as a subgraph, $G$ is generically 2-rigid.

Conversely, if $G$ is generically 2-rigid, it has a spanning subgraph $G'$ which is generically 2-isostatic, and which is thus representable as a proper $3T2$-graph $G' = \cup T_i'$. Each end of each edge $e$ in the complementary subgraph $G \backslash G'$ is incident with two of these three trees, so the ends of $e$ are in a common tree $T_i$. Add $e$ to that tree, and do likewise for every edge $e$ in $G \backslash G'$, enlarging the three subgraphs $T_i$ to subsets $G_i$ whose edge-disjoint union is all of $G$. These three subgraphs are connected, and satisfy the conditions given in the statement of the theorem. □

## 5. The algorithm

If $J$ and $K$ are disjoint independent subsets of the edge set $E$ of a graph $G = (V, E)$, the nest of the ordered pair $J, K$ is the following sequence of closed subsets of $E$:

$$\begin{gathered} \textit{nest}\ (J,K) = S_0, S_1, \ldots, \quad \text{where} \\ S_0 = E,\ S_1 = \langle J \cap S_0 \rangle,\ S_2 = \langle K \cap S_1 \rangle, \\ S_3 = \langle J \cap S_2 \rangle,\ S_4 = \langle K \cap S_3 \rangle,\ \ldots\ . \end{gathered}$$

It is clear that the sets $S_i$ are "nested", that is: $S_i \subseteq S_{i-1}$ for all $i \geq 1$, because $S_i$ is the closure of a subset of the closed set $S_{i-1}$. (Each successive closed set $S_i$ in the nest is that part of the previous set $S_{i-1}$ that is spanned from within by edges of the other independent set.) The *label* of an edge $e$ is equal to $i$ if $e \in S_i \backslash S_{i+1}$, and is "$\infty$" if $e \in S_i$ for all $i = 0, 1, \ldots$.

**Proposition 8.** *In the nest of a pair $J, K$ of disjoint independent sets, the (finite) labels of edges in $J$ are odd, those of edges in $K$ are even.*

Proof: If $\lambda(e) = i$, with $i$ even, then $e \in S_i$, but $e \notin S_{i+1} = \langle S_i \cap J \rangle$. In particular, $e \notin S_i \cap J$, so $e \notin J$. So $\lambda(e)$ is odd for all $e \in J$. Similarly, $\lambda(e)$ is even for all $e \in K$. □

**Example.** The graph in Figure 2 is a typical intermediate stage in the application of the algorithm we intend to describe. The two independent sets $J$ (wide black lines) and $K$ (wide gray lines) cannot be extended simply by adding an edge ($eg$ or $hi$) to $J$ or to $K$. In the figure at the right, we indicate the label $\lambda(e)$ for each edge $e$. The closed set $S_1$ contains those edges labelled "1" through "5", that is, all but six edges of the graph. Sets $S_0$ through $S_5$ are non-empty. For indices $i \geq 6$, $S_i$ is empty.

The idea to construct the nest of a family of disjoint independent sets is the essence of Edmonds' algorithm for minimum partition into independent sets. The nest provides a finite limit (the rank of the graph) for those chains of substitutions which add one new edge to one independent set. As we shall see, it is also the key to the detection of generically over-braced subframeworks in graphs with $2v - 3$ edges on $v$ vertices.

**Proposition 9.** *Let nest $(J, K) = \{S_i\}$ be the nest of a pair $J, K$ of disjoint independent sets of edges of a generically 2-isostatic graph $G$. Then for some index $i \leq r(G)$, $S_i$ is empty.*

Proof: It is possible that $S_0 = S_1$. This happens if the set $J$ is a spanning tree in $G$. But for $i \geq 1$, and so long as $S_i$ is not empty, $r(S_{i+1}) < r(S_i)$. To see why, assume without loss of generality that $i$ is even, so $S_{i+1} = \langle J \cap S_i \rangle$. If $r(S_{i+1}) = r(S_i)$, then $S_{i+1} = S_i$ (because $S_{i+1} \subseteq S_i$ and both sets are closed). Furthermore, $J \cap S_i$ spans $S_{i+1}$, just as $K \cap S_{i-1}$ spans $S_i$. Let $R$ be any connected component of $S_i$. Then $J \cap R$ and $K \cap R$ are trees which span $R$. This cannot happen in a generically 2-isostatic graph, so $r(S_{i+1}) < r(S_i)$ whenever $S_i$ is not empty, for $i \geq 1$. □

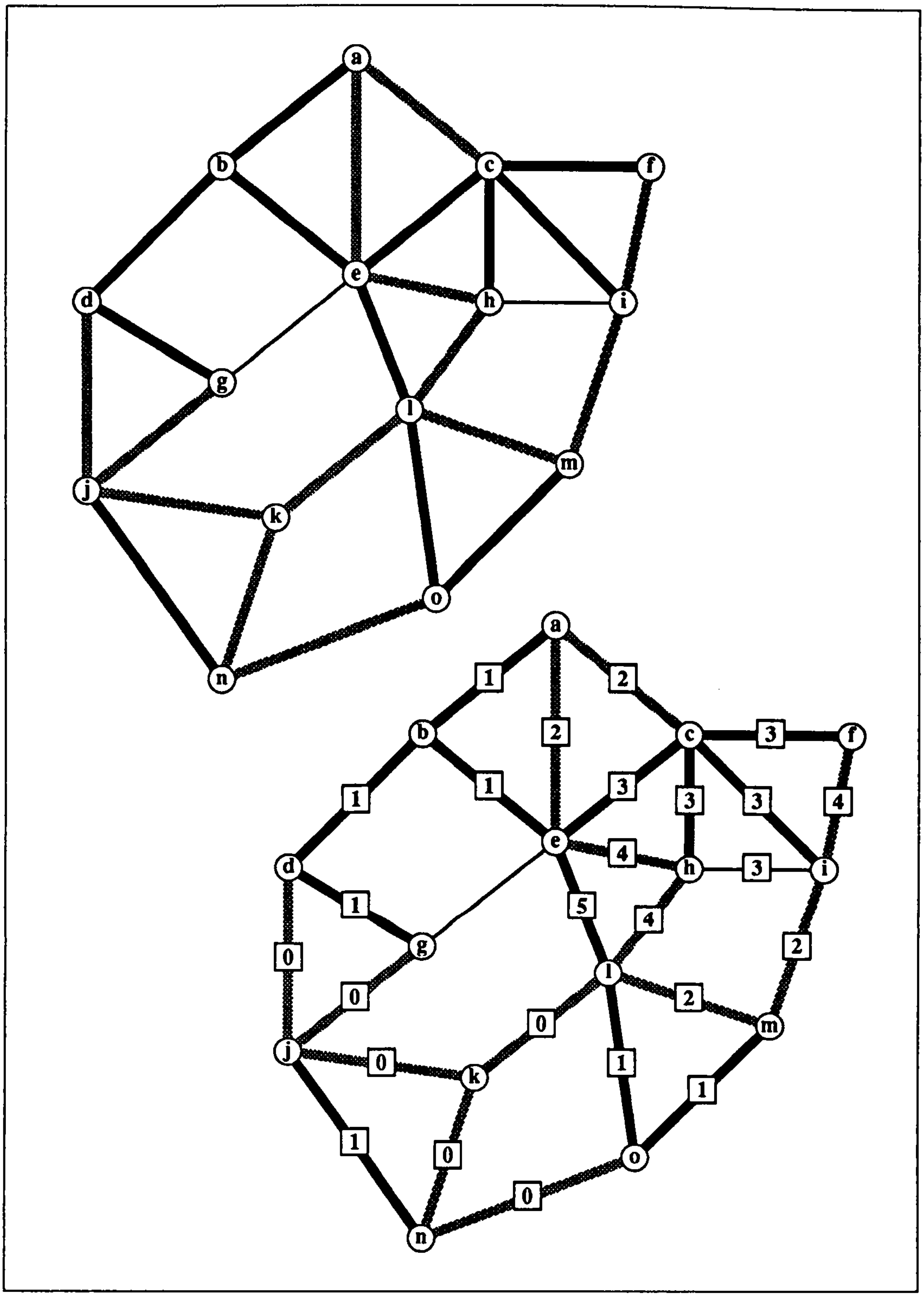


Figure 2. The nest of a pair of independent sets: J (black), K (gray)

**Proposition 10.** *Let nest $(J,K)=\{S_i\}$ be the nest of a pair $J,K$ of disjoint independent sets of edges in a graph $G$. For any even-labelled edge $x$, either $x \notin \langle J\rangle$, or the unique cycle $C \subset J \cup x$ contains an odd-labelled edge $y$ with $\lambda(y) < \lambda(x)$. For any odd-labelled edge $x$, either $x \notin \langle K\rangle$, or the unique cycle $C \subset K \cup x$ contains an even-labelled edge $y$ with $\lambda(y) < \lambda(x)$.*

Proof: Without loss of generality, assume $\lambda(x)$ is even. If $x \in \langle J\rangle$ and if all edges $y \neq x$ in the unique cycle $C \subset J \cup x$ have $\lambda(y) \geq i$, then $x$ is in the closure $\langle J \cap S_i\rangle = S_{i+1}$, contradicting the hypothesis. □

Propositions 9 and 10 demonstrate the possibility of a finite chain of substitutions which will permit the enlargement of any pair of disjoint independent sets whose union is not already the entire graph $G$. This procedure appears at step 6 in the following algorithm.

**Algorithm 11.** *To determine whether a graph $G=(V,E)$ is generically 2-isostatic.*

*0. Say $G$ has $v$ vertices. If $G$ is not connected, or if it does not have exactly $2v-3$ edges, it is not generically 2-isostatic. Stop. Otherwise, continue.*

*1. As initial values for $J$ and $K$, independent subsets of $E$, set both equal to the empty set.*

*2. If $J$ and $K$ are both non-empty, compute nest $(J,K)$. If, at some point, $S_{i+1}=S_i \neq \emptyset$, the graph $G$ is not generically 2-isostatic. Stop. Otherwise, the construction of the nest terminates with $S_i \neq \emptyset$, $S_{i+1}=\emptyset$, for some index $i$.*

*3. If $J \cup K = E$, then $G$ is generically 2-isostatic. Stop. Otherwise, let $x$ be an edge of $G$ which is not in $J \cup K$.*

*4. If $x \notin \langle J\rangle$, replace $J$ by $J \cup x$, and go to step 2.*

*5. If $x \notin \langle K\rangle$, replace $K$ by $K \cup x$, and go to step 2.*

*6. Say $\lambda(x) = i$. By Proposition 10, the unique cycle $C$,*

$$C \subseteq J \cup x \text{ if } i \text{ is even,}$$
$$C \subseteq K \cup x \text{ if } i \text{ is odd,}$$

*contains an edge $y$ with $\lambda(y) < i$, of opposite parity. Replace*

$$J \text{ by } J \cup x - y \text{ if } i \text{ is even,}$$
$$K \text{ by } K \cup x - y \text{ if } i \text{ is odd,}$$

*then replace $x$ by $y$, and go to step 4.*

Proof of the validity of this algorithm:

(a) If, in the construction of *nest* $(J, K)$ in step 2, we find $S_{i+1} = S_i \neq \emptyset$, then the subgraph $S_i$ is spanned by subgraphs of the independent sets $J$ and $K$. Any connected component of $S_i$, say with $v'$ vertices, will have at least $2v' - 2$ edges, so the graph $G$ does not satisfy the conditions of Laman's theorem.

(b) If the construction of *nest* $(J, K)$ does not start to "cycle", with $S_{i+1} = S_i \neq \emptyset$, it will terminate in no more than $r(G)$ steps, by proposition 9.

(c) Say $J \cup K = E$ in step 3. Since $J$ and $K$ are independent sets, they contain no more than $v - 1$ edges. Since $J \cup K = E$, they have $2v - 3$ edges in all. Consequently, one of them (say $J$) has $v - 1$ edges, and the other ($K$) has $v - 2$ edges. Thus $J$ is a spanning tree, connecting all vertices of $G$, while $K$ joins the vertex set $V$ into two connected components. (One such component may of course be a single vertex.) Let $T_1$ be the tree $J$, $T_2$ and $T_3$ the two connected components relative to the independent set $K$. Then $\{T_i\}$ is a $3T2$-structure on the graph $G$. If it were not proper, there would be a subset $V'$ of $G$ with at least two vertices, spanned by subtrees of $T_1$ and $T_2$, or of $T_1$ and $T_3$. (Trees $T_2$ and $T_3$ have no vertices in common.) But then the induced subgraph $G|_{V'}$ will be in all sets $S_i$ of *nest* $(J, K)$ , and the construction of the nest would not have succeeded, at step 2. So $(G, T_i)$ is a proper $3T2$-graph, and is generically isostatic.

(d) If an edge is added to $J$ at step 4 or to $K$ at step 5, the new subsets $J$ and $K$ are still disjoint and independent.

(e) When edge $y$ replaces edge $x$ in $J$, step 6, the set $J \cup y - x$ is also independent, because $y$ is in the unique cycle $C$ in $J \cup x$. If the new element "$x$"

is not added to $J$ or to $K$ in step 4 or 5, the value of the index "$i$" at the next passage through step 6 will decrease, because this new "$x$" (formerly "$y$") is not in $S_i$. Consequently, at most $r(G)$ passes through step 6 will be necessary before an edge is added in step 4 or 5, and we reenter steps 2 and 3.

(f) If the algorithm is not terminated at step 0 or 2 with the information that $G$ is not generically 2-isostatic, then at the $(2v-2)^{nd}$ entry into step 3, $J \cup K = E$, and a proper $3T2$-structure has been constructed. The algorithm is valid. □

Note that all $3T2$-structures produced by this algorithm have the property that one of the three trees spans the entire graph. More general $3T2$-structures, such as those in Figure 1 (A and C), cannot arise in this way.

## 6. Improvements to the algorithm

Since the most time-consuming aspect of the program is the computation of the nest of two independent sets, it would be better to simplify this process, and to postpone it until it is really necessary.

First, consider postponing the nest computation. An appropriate strategy is, at step 1, to build the independent sets $J$ and $K$ by a greedy algorithm, for as long as possible. Say we were to go through the set of edges of $G$, adding each edge $x$ to $J$ if $x \notin \langle J \rangle$, to $K$ if $x \in \langle J \rangle \backslash \langle K \rangle$, otherwise saving $x$ for later processing. We arrive in this way at a situation where $J$ is a spanning tree for $G$, and for every $x \notin J \cup K$, $x \in \langle J \rangle \cap \langle K \rangle$. (On the average, there are very few such edges.) At this stage, the sets $J$ and $K$ may already contain trees with the same span, but this will be discovered the first time we compute the nest. If $J \cup K = E$, we perform this test immediately. Otherwise, we will compute the nest once for each edge remaining to be added.

One basic step, which is repeated many times during any single computation of *nest* $(J, K)$, is to decide

**Step.** For a given edge $e \in E$, and a given subset $S \subseteq E$, whether $e \in \langle S \cap J \rangle$ (or whether $e \in \langle S \cap K \rangle$).

This step is particularly time-consuming if it is not supported by the provision of an appropriate data structure. With this in mind, let $J_{ab}$ be the (unique) path from a vertex $a$ to a vertex $b$ in an independent set $J$, and set $J_{ab} = \emptyset$ if $a$ and $b$ are not connected by $J$. What is required is an updated list of paths $J_{ab}$, $K_{ab}$ for every edge $e = ab$ in $G$. The above "step" is simplified by the observation that

$$e \in \langle S \cap J \rangle \text{ if and only if } \emptyset \neq J_{ab} \subseteq S.$$

It is no longer necessary to "search" through the set $S \cap J$ for a connection from $a$ to $b$.

Fortunately, the list $L = \{(J_{ab}, K_{ab}), \text{ for all } e = ab \text{ in } G\}$ is relatively easy to update after each substitution of one edge for another in $J$ or in $K$, or after the addition of an edge to $J$ or to $K$. For instance, in Figure 3(A), when the edge $ab$ is substituted for the edge $cd$, the path $J_{yq} = yxdcsrq$ is changed to $J'_{yq} = yxzbarq$. This can be accomplished in two steps. First, replace "$dc$" in $J_{yq}$ by two segments $dxzb$ and $arsc$ of the former path $J_{ab} = arscdxzb$. This produces a path

$$yxdxzbarscsrq$$

with two palindromes $xdx$ and $rscsr$ which can be removed to yield $yxzbarq$. Such calculations are straightforward, not requiring search in the graph, and thus are relatively immune to "combinatorial explosion".

If an edge $e = ab$ is added to an independent set $J$, where $e \notin \langle J \rangle$, the paths $J_{xy}$ need be updated only when $J_{xy} = \emptyset$, and there are $J$-paths from $x$ and $y$ to $a$ and $b$ (in some order). In that case, the new path $J'_{xy}$ is the concatenation of $J_{xa}$ with $J_{by}$ or of $J_{xb}$ with $J_{ay}$, as the case may be. (In only one of these two cases will the required paths be non-empty.) See Figure 3(B).

With an updated pathlist $L$ available, the nest is calculated in an orderly fashion. Take $S = E$ and label all edges "0". At each pass $i$, say for $i$ odd, we look down the list of $J$-paths for edges $ab$ whose labels are equal to $i - 1$. If $J_{ab}$ is non-empty, and if every edge $cd$ in $J_{ab}$ has label $\geq i - 1$, then we increment the label of the edge $ab$ to $i$.

If, at the $i^{th}$ pass, the entire (non-empty) set of labels $i - 1$ are incremented to $i$, we have demonstrated the existence of a subgraph spanned by subtrees of $J$ and $K$. If, at the $i^{th}$ pass, no label is incremented, the nest is complete.

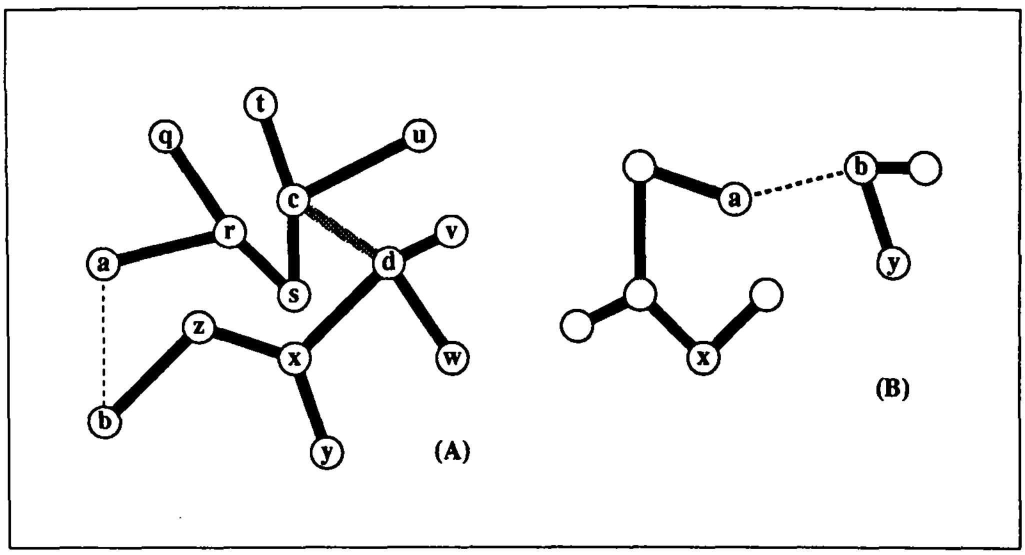


Figure 3. Updating of tree-paths.

The above process is normally carried out in order to add one particular edge $e = ab$ to the union $J \cup K$. In that event, the computation of the nest need only be carried out until a pass is completed in which the label of the edge $e$ is not incremented. Subsequent exchanges will involve only edges for which the definitive labels are already known.

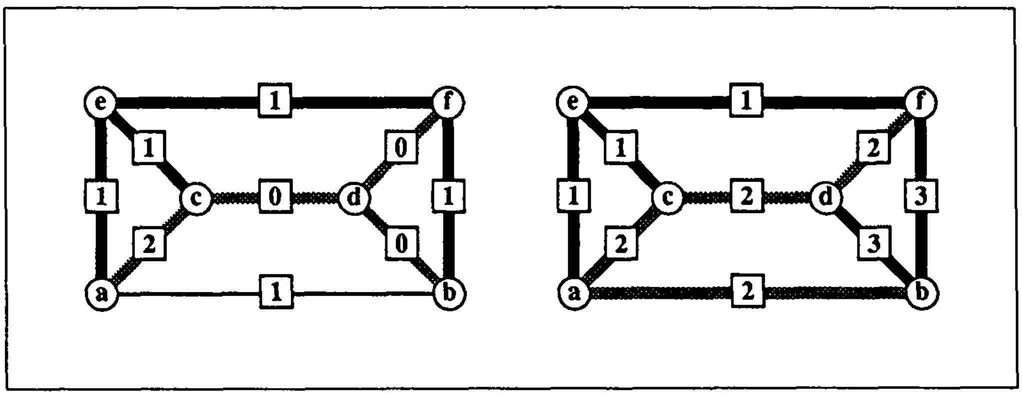


Figure 4. Updating of labels in a nest.

It is tempting to try directly to update the list of labels of edges, rather than have recourse to the list $L$ of $J$-paths and $K$-paths. At present, this seems not to be possible. That is, the updated labels seem not to depend simply on the labels of those edges involved in the exchange. In Figure 4, the edge $ab$ is added to $K$, in exchange for the edge $bd$, which is shifted to $J$. How can we conclude simply that the label on edge $df$ should be augmented from 0 to 2, while that that on edge $bf$ should be increased from 1 to 3?

## 7. Henneberg sequences

A graph with $n$ vertices has a *Henneberg 2-sequence* if and only of there is a sequence $G_2, G_3, \dots, G_n = G$ of graphs, beginning with the graph $G_2$ which consists of a single edge joining two vertices, such that each successive graph $G_i$ is obtained from its predecessor $G_{i-1}$ by adding a two-valent vertex, or by deleting an edge $ab$ and inserting a three-valent vertex $q$ with edges $qa$, $qb$, and $qc$, for some vertex $c$ distinct from $a$ and $b$.

Tay and Whiteley [T1] proved that if a graph is generically 2-isostatic, it has a Henneberg 2-sequence (and conversely). Their proof uses mechanical properties of subframeworks of isostatic frameworks. Given a Henneberg sequence, is there a way to construct a corresponding $3T2$-structure, and conversely? The following algorithm, due to Walter Whiteley (unpublished work), shows that the answer to the first part of this question is affirmative. It builds a $3T2$-structure in which the tree $T_1$ is a spanning tree.

**Algorithm 12.** *Given a Henneberg sequence of graphs $G_i$ on $i$ vertices, for $i = 2, 3, \dots, n$, with $G_n = G$:*

*1. Set $i = 2$. Let $T_1$ be the single edge of the graph $G_2$. Let $T_2$ and $T_3$ be the single-vertex trees at the ends of the edge, as in Figure 1(D).*

*2. If $i = n$, a proper $3T2$-structure has been obtained for $G$; Stop. Otherwise, increment $i$ to $i + 1$.*

*3. If graph $G_{i+1}$ is obtained by adding a 2-valent vertex $q$ with edges from $q$ to* *vertices $a, b$ of $G_i$, then add the edge $aq$ to tree $T_1$, the edge $bq$ to the other tree ($T_2$ or $T_3$) incident with vertex $b$. Go to step 2.*

*4. Otherwise, the graph $G_{i+1}$ is obtained by splitting an edge $ab$ by a new vertex $q$, making it 3-valent by adding an edge $ac$ to some vertex $c$ ($c \neq a, c \neq b$) of $G_i$. Add the edges $aq$ and $bq$ to the tree which contained the split edge $ab$. The third edge $cq$ is added to tree $T_1$ if the edge $ab$ was in $T_2$ or $T_3$. Otherwise, the edge $ab$ was in $T_1$, and the third edge $cq$ is added to the other tree ($T_2$ or $T_3$) present at vertex $c$. Go to step 2.*

Proof of the validity of this algorithm: The result at stage $n$ is a $3T2$-structure on $G$. It is proper, because any graph with a Henneberg sequence is generically isostatic, and any $3T2$-structure on an isostatic graph is proper. □

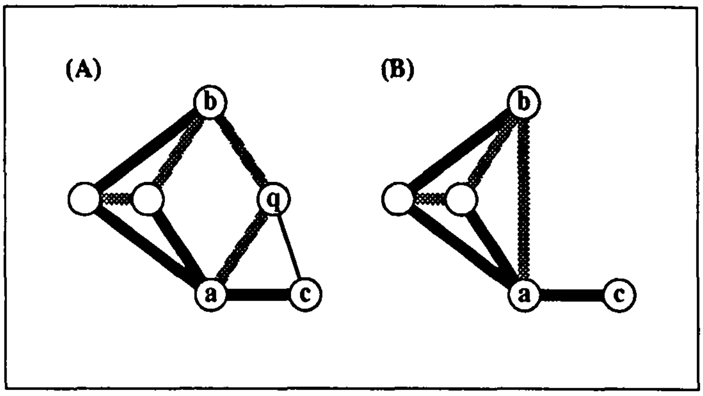


Figure 5. The 3T2-structure is not a reliable guide to development of a Henneberg sequence.

The opposite passage, from $3T2$-structures to Henneberg sequences, is not so easy. Say we try to reverse the procedures used in Algorithm 12. In particular, we try to use the rule provided by Step 4 to predict which edge to place between two of the vertices $a, b, c$ when a 3-valent vertex $q$ is removed. If we use this choice in Figure 5, we would place an edge between vertices $a$ and $b$, But Figure 5(B) is not a proper $3T2$-structure. This difficulty was not present in Algorithm 12, because every graph with a Henneberg sequence is generically 2-isostatic, and every $3T2$-structure on such a graph is proper. To bypass this difficulty, we will have to consider implicit edges.

## 8. Implicit edges

In a graph $G$, regarded as a generic bar and joint framework in the plane, we say a pair $ab$ of vertices is generically 2-implicit (as an edge) in $G$ if and only if the load "equal and opposite forces applied at $a$ and $b$, along the line $ab$" is resolvable by the framework $G$ in general position. Equivalently, the pair $ab$ is generically 2-implicit as an edge if and only if those vertices remain initially at a constant distance from one another during any infinitesimal (equivalently: real mechanical) motion of the framework $G$, starting in a general position. The correct determination of implicit edges is crucial to many aspects of framework analysis and synthesis. A significant theoretical objective is to obtain a purely combinatorial characterization of implicit edges of generic frameworks. This problem remains intransigent for structures in 3-space, but for 2-generic frameworks, the above algorithmic approach provides a practical solution.

Again, rather than deal directly with the mechanical properties of graphs in general position, we appeal to Laman's Theorem for a combinatorial characterization of implicit edges. Recall that by Laman's Theorem, a framework is generically 2-independent if and only if for every subset $V'$ of $v' \geq 2$ of its vertices, the induced subgraph $G|_{V'}$ has no more than $2v' - 3$ edges. A *generic 2-circuit* is a minimal graph which fails to be generically 2-independent. That is, it has $2v - 2$ edges on $v$ vertices, and every proper subgraph on $v'$ vertices, for $v' \geq 2$, has $\leq 2v' - 3$ edges. Finally, an edge $e = ab$ is *generically 2-implicit* in a graph $G$ if and only if there is a generically 2-isostatic subgraph $G' \subseteq G$ such that $G' + e$ is a generic 2-circuit.

A $2T2$-graph is simply the edge-disjoint union of two spanning trees. For the purposes of this discussion, we say a $2T2$-graph $G$ is *proper* if no proper subgraph of $G$ (with at least one edge) is spanned by subtrees of those two trees.

**Proposition 13.** *A graph $G$ is a generic 2-circuit if and only if it has a representation as a proper $2T2$ graph.*

Proof: A minimal graph which violates Laman's conditions for generic 2-independence will have exactly $2v - 2$ edges on $v$ vertices, and no more than $2v' - 3$ edges on any proper subset of $v'$ of those vertices. By Tutte's theorem [T2] this is equivalent to the statement that $G$ is the edge-disjoint union of two spanning trees $T_i$. If some proper subgraph $G'$ on $v'$ vertices were also cospanned by two trees, then $G'$ would have $2v' - 2$ edges. The intersections of $T_1$ and $T_2$ with $G'$ would be independent sets, each with $\leq 2v' - 1$ edges. Since they cover $G'$, they must both have exactly $2v' - 1$ edges, and must be spanning trees. So $G'$ is cospanned by subtrees of $T_1$ and $T_2$. We see that "minimal $2T2$" is equivalent to "proper $2T2$". □

**Algorithm 14.** *Given a graph $G = (V, E)$ and a pair $ab$ of vertices of $G$, to decide whether the pair $ab$ is generically 2-implicit as an edge of $G$:*

*Part A.*

*1. As initial values for $J$ and $K$, independent subsets of $G$, set both equal to the empty set.*

*2. If $J$ and $K$ are both non-empty, compute nest $(J, K)$. If, at some point, $S_{i+1} = S_i \neq \emptyset$, delete an edge of $S_i$ from $G$ (and from $J$ or $K$, as appropriate).*

*3. If $J \cup K = E$, $G$ is a generically 2-independent graph, while $J, K$ partition the edge set of the (now reduced) graph $G$; Go to Part B. Otherwise, let $x$ be an edge of $G$ which is not in $J \cup K$.*

*4. If $x \notin \langle J \rangle$, replace $J$ by $J \cup x$, and go to step 2.*

*5. If $x \notin \langle K \rangle$, replace $K$ by $K \cup x$, and go to step 2.*

*6. Say $\lambda(x) = i$. By Proposition 10, the unique cycle $C$,*

$$C \subseteq J \cup x \quad \text{if } i \text{ is even,}$$
$$C \subseteq K \cup x \quad \text{if } i \text{ is odd,}$$

*contains an edge $y$ with $\lambda(y) < i$, of opposite parity. Replace*

$$J \text{ by } J \cup x - y \text{ if } i \text{ is even,}$$
$$K \text{ by } K \cup x - y \text{ if } i \text{ is odd,}$$

*then replace $x$ by $y$, and go to step 4.*

*Part B.*

1. *Let $x = ab$. Add the edge $x$ to $G$.*

2. *Compute the nest $(J, K)$. The construction of the nest terminates with $S_i \neq \emptyset$, $S_{i+1} = \emptyset$, for some index $i$.*

3. *If $x \notin \langle J \rangle$, replace $J$ by $J \cup x$, and go to step 6.*

4. *If $x \notin \langle K \rangle$, replace $K$ by $K \cup x$, go to step 6.*

5. *Say $\lambda(x) = i$. By Proposition 10, the unique cycle $C$,*

$$C \subseteq J \cup x \quad \text{if } i \text{ is even,}$$
$$C \subseteq K \cup x \quad \text{if } i \text{ is odd,}$$

*contains an edge $y$ with $\lambda(y) < i$, of opposite parity. Replace*

$$J \quad \text{by } J \cup x - y \text{ if } i \text{ is even,}$$
$$K \quad \text{by } K \cup x - y \text{ if } i \text{ is odd,}$$

*then replace $x$ by $y$, and go to step 3.*

6. *Compute nest $(J, K)$. If, at some point, $S_{i+1} = S_i \neq \emptyset$, the pair $ab$ is generically 2-implicit as an edge of the original framework $G$. If not, the construction of the nest terminates with $S_i \neq \emptyset, S_{i+1} = \emptyset$ for some index $i$, and the pair $ab$ is not generically 2-implicit in $G$.*

Proof of the validity of algorithm 14: This algorithm differs from algorithm 11 only in two respects. In part $A$ of the present algorithm, whenever we discover a generic 2-dependence in the graph $G$, we remove an edge of the 2-dependent subgraph. So we will eventually find $J \cup K = E$ in step 3, and $G$ will have been reduced to a generically 2-independent graph.

If $G$ is generically 2-independent, it remains only to determine whether the addition of the pair $ab$ as an edge of $G$ introduces a generic 2-dependence. The process of adjoining the edge $ab$ to one of the independent sets $J$ or $K$, after a certain number of exchanges, will invariably succeed (step B2). (It would take $2v' - 1$ edges on some set of $v'$ vertices to block this process.) The edge $ab$ forms a generic 2-circuit with a part of the original graph $G$ if and only if the final calculation of the *nest* $(J, K)$ "cycles" on a non-empty set of edges cospanned by independent subsets of $J$ and $K$. □

**Algorithm 15.** *To determine a Henneberg 2-sequence for the construction of a generically 2-isostatic graph $G$.*

*(a) Set $J = K = F$. Use algorithm 11 to find a 3T2 structure for $G$.*

*(b) Set $G_0 = G$.*

*(c) If $G_0$ is a single edge, the Henneberg sequence is complete. Stop.*

*(d) For all graphs $G_i$ which have been defined, relabel them $G_{i+1}$.*

*(e) If $G$ has a two-valent vertex, say $q$, with incident edges, say $aq$, $bq$, then replace $V$ by $V - q$ and $E$ by $E \backslash - \{aq, bq\}$. Drop the edges $aq$ and $bq$ from the independent sets $J$ and $K$. Go to step (b).*

*(f) $G$ has a three-valent vertex, say $q$, with incident edges, say $aq$, $bq$, $cq$. Delete that vertex and those edges from $G$, and also from the independent sets $J$ and $K$. Store the current values of $J$ and $K$ as $J'$ and $K'$.*

*(g) Letting edge $x$ take successive values $ab$, $ac$, $bc$, do the following: Add edge $x$ to the graph $G$. Use algorithm 14 (starting with $J = J'$, $DK = K'$) to determine whether the pair $x$ is generically 2-implicit in $G$. If so, delete the edge $x$ from $G$ and restart this step (g) with the next edge ($ac$ or $bc$). Otherwise, go to step (b).*
□

We do not have to recompute the independent sets $J$ and $K$ at every reduction of the graph $G$. The restrictions of the previous sets $J$ and $K$ to the reduced graph are still independent. Indeed, when a two-valent vertex is removed, the restrictions of $J$ and $K$ are already a 3T2 structure. When a three-valent vertex is replaced by an edge, the sets $J$ and $K$ resulting from the test (that the edge is not implicit) provide a $3T2$ structure for the reduced graph. Furthermore, during the tests for implicit edges, only part (B) of algorithm 14 comes into play, and it uses only two computations of the nest. By the Tay-Whiteley theorem, if the original graph is generically 2-isostatic, there is always a correct replacement available, so this approach will always produce a verified Henneberg 2-sequence.

## 9. Frameworks in 3-space

The multiple-tree partition approach might conceivably make some contribution to the still difficult problem of identifying generically isostatic frameworks in 3-space. The corresponding combinatorial structures will be $6T3$ (as we show below), that is, with a partition of the edge set of a graph into 6 trees, exactly 3 of which meet at each vertex. The triangle and the complete graph on 4 vertices are two simple examples of such graphs.

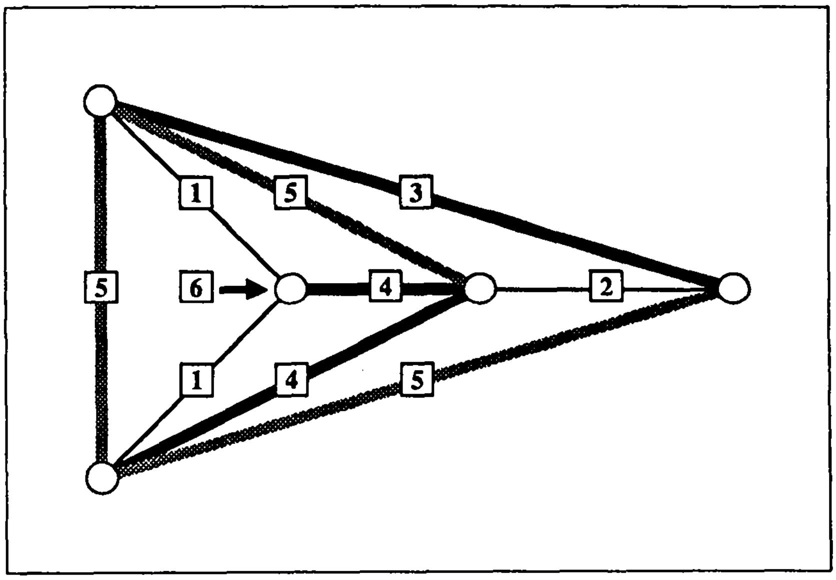


**Figure 6.** A generically 3-isostatic graph, with a proper 6T3 structure

The straightforward analogue of Laman's conditions are necessary but not sufficient as a description of generically 3-isostatic graphs. Such graphs have $3v-6$

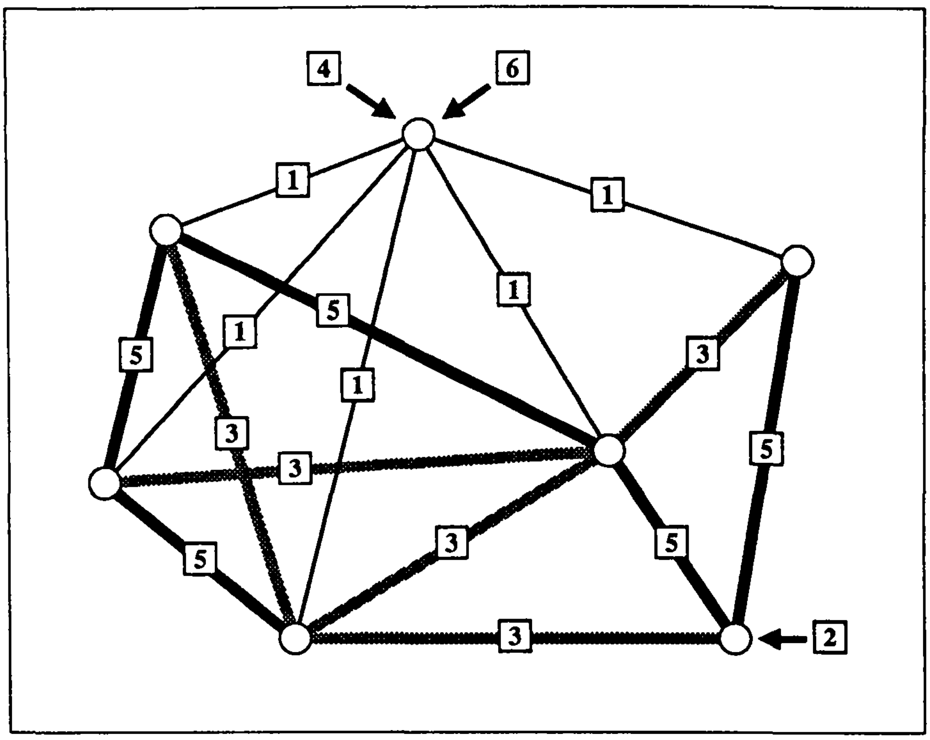


Figure 7. A generically 3-dependent, flexible graph, with an improper 6T3 structure

edges on $v$ vertices, and no more than $3v' - 6$ edges on any subgraph with $v'$ vertices, for $3 \leq v' < v$. Consider the graph obtained by taking two copies of the complete graph on 5 vertices and joining them along an edge, then removing the common edge. This graph, commonly called the framework of "two bananas", has $3v - 6$ edges, and no over-braced proper subgraph, but is not generically 3-isostatic. The two "bananas" can rotate relative to one another along the former common edge. A strengthened generalization of the notion of "proper" is certainly necessary, but it has not yet been found. Whether this will be easier to express using tree partitions than it is in conventional attempts to generalize Laman's theorem, remains to be seen.

**Proposition 16.** *Any generically d-isostatic graph $G = (V, E)$ is representable as an $nTd$- graph, where $n$ is the binomial coefficient $\binom{d+1}{2}$, that is, $G$ is representable as the edge-disjoint union of $n$ trees, exactly $d$ of which meet at every vertex.*

Proof: Let $n = \binom{n+1}{2}$. Generically $d$-isostatic graphs have $dv - n$ edges on $v$ vertices, and no more than $dv' - n$ edges on any subgraph with $v'$ edges, for $v' \geq d$. Since a generically $d$-isostatic graph never has more than $dv' - d$ edges on any subset of $v' \geq d$ of its vertices, it can be represented as the edge-disjoint union of $d$ independent sets $I_i$, by Tutte's theorem [$T_2$]. Say these sets $I_i$ have $e_i$ edges and thus connect the entire vertex set $V$ into $v - e_i$ components. Since $dv - n = \sum e_i$, we know that $\sum(v - e_i) = dv - \sum e_i = n$. The total number of connected components (trees) in these three subgraphs on the set $V$ of vertices is $n$. Every vertex is in exactly $d$ of these trees, one from each independent set $I_i$. □

In particular, any generically 3-isostatic graph is representable as a $6T3$-graph, that is, as the edge-disjoint union of six trees, exactly three of which meet at every vertex. Some of the trees may consist of single vertices, and these vertices need not be distinct! Figure 6-Figure 8 illustrate such $6T3$-graphs. Figure 6 shows a framework that is generically 3-isostatic. The framework in Figure 7 has $3v - 6$ edges on $v$ vertices, and has an overbraced subframework. Figure 8 shows a framework with "hinges", which is not generically 3-isostatic, and has no violation of the "$3v' - 6$" count. The decompositions into three independent sets are shown in different tones of lines. The six components of these three sets are indicated by numerical labels. When there is a subframework with more than $3v' - 6$ edges on a set $V'$ of $v'$ vertices, as in Figure 7, the three induced subgraphs $I_i|_{V'}$, each on the entire set $V'$ of edges, have a total of at most five connected components.

The framework in Figure 8 is formed from a chain of four copies of the complete graph $K_5$ on 5 vertices, joined along edges (which are then removed). Without the two outer edges (indicated by arcs), the graph is a generic 3-circuit, with $34 = 3v - 8$ edges and three internal degrees of freedom, namely the rotations about the three "hinges". We add two more edges to bring the count up to $36 = 3v - 6$, and remove two of the three internal degrees of freedom. The framework still has one internal degree of freedom, so it is not generically 3-isostatic. With the two additional edges in place, the existence of "hinges" is no longer recoverable from the usual topological considerations, such as the decomposition of the graph into 3-connected components. It remains to be seen whether frameworks with "hinges" can be identified by some property of their $6T3$ structures.

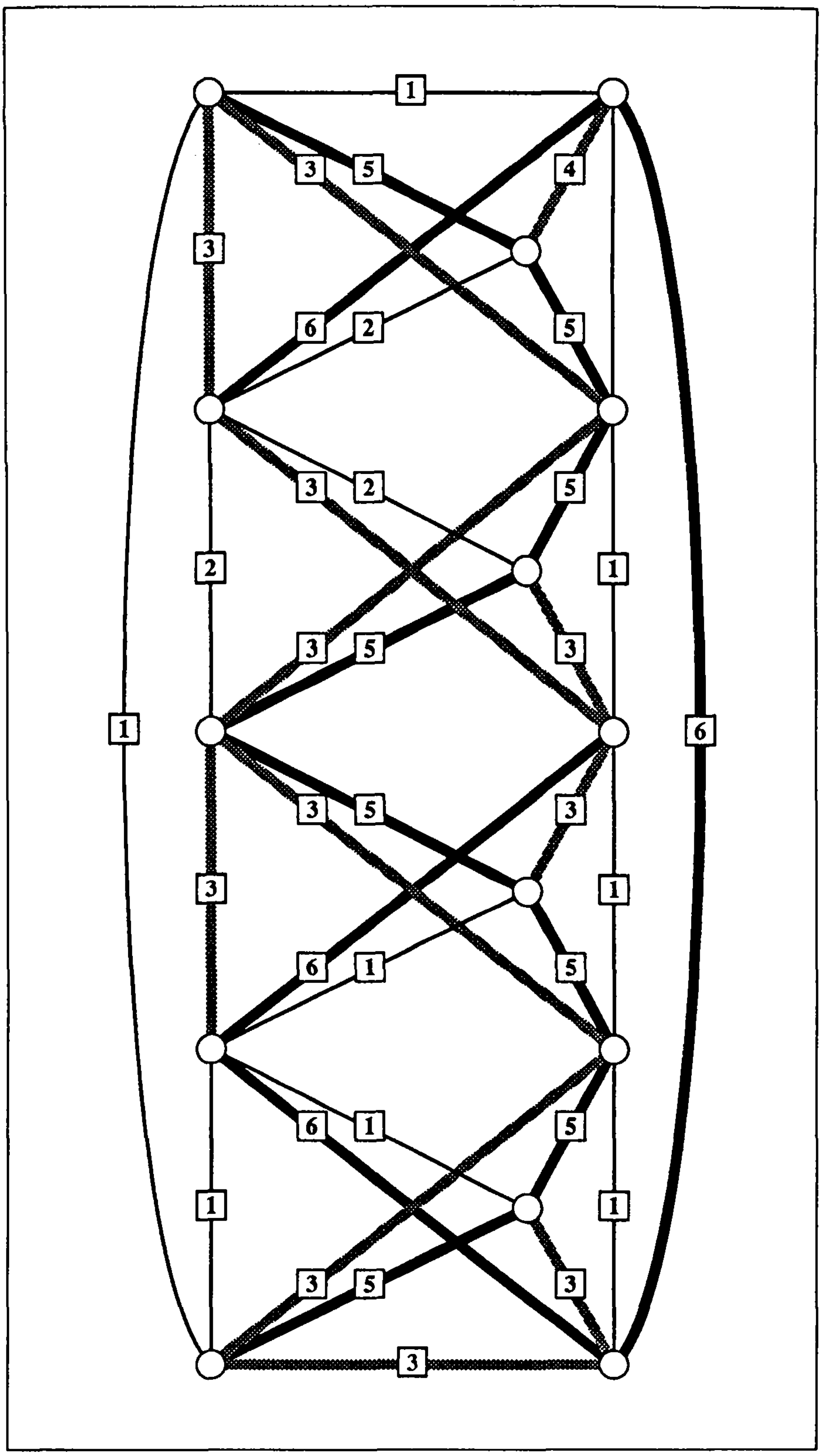


Figure 8. A generically dependent graph in 3-space, with a proper 6T3 structure

## 10. Related work

The referees for this article kindly led us to some related work on plane frameworks, independent of and concurrent with our work on $3T2$ structures. In their paper *Forests, Frames and Games: Algorithms for Matroid Sums and Applications* [G&W], Gabow and Westermann likewise suggested using Edmonds' algorithm to detect 2-generically isostatic frameworks. Their research yields essentially the same algorithm. Edmonds' algorithm is there used first to find a spanning tree and a disjoint two-component forest, and then to test that there is no over-braced $(2v' - 2)$-subgraph. Their treatment of plane frameworks is immersed in a rather complex chain of improvements to, and varied applications of, matroid partition algorithms. Let's take a look at what they propose, in the present context.

Their article deals with arbitrary matroid sums, but for the applications to generic rigidity of plane frameworks, we restrict our attention to the matroid $M = G^2$, the matroid sum of the graphic matroid $G$ with itself. In this matroid, a set of edges is independent if and only if it is the edge-disjoint union of two forests in $G$. A *clump* in $M$, in their terminology, is a set $L$ of edges of $G$ that can be partitioned into forests $L_1, L_2$ such that $L$ is cospanned by $L_1$ and $L_2$:

$$L \subseteq \langle L_1 \rangle \cap \langle L_2 \rangle.$$

There is a non-empty clump $L \subseteq E$ of $G^2$ in the edge set of a graph $G$ if and only if there is a set of nodes cospanned by disjoint trees.

Their claims for increased efficiency are based on three algorithmic devices: "breadth-first scanning", "cyclic scanning", and "batch" construction of augmenting paths. In breadth-first scanning, they reduce the time required to build augmenting paths by working "modulo" previously-determined clumps. Cyclic scanning is an efficient method to select forests in which to look for the next step in an augmenting path. The batch mode economises on the information collection used in branching to find augmenting paths.

None of these devices have any real bearing on the present algorithm. We are working with only two forests, so cyclic scanning is the same as breadth-first scanning. There is no need to work modulo a clump, following an unsuccessful attempt to find an augmenting path. We simply stop, knowing that the presence of a set of vertices cospanned by two edge-disjoint trees means that the graph is not isostatic. The batch mode is advantageous only when one must search for a good edge from which to start an augmenting path. In our algorithm, we take advantage of the fact that the construction of augmenting paths is "greedy": for any pair of edge-disjoint forests and for any additional edge in a generically 2-isostatic framework, there is an augmenting path which adds that edge to one of the two forests. Any failure to construct an augmenting path from a given element in a general graph means simply that the graph is not isostatic.

Encouraged by the success of Gabow and Westermann in improving the Edmonds algorithm for arbitrary matroid sums, we have taken the opportunity to add section 6, above, in which we update the nest, rather than computing it from "scratch", after successfully adding an edge to the pair of forests.

## 11. Suggestions for further work

(1) Algorithm 11 produces only proper $3T2$-graphs $(G, T_i)$ in which one of the trees $T_i$ spans the graph $G$. A modified algorithm which deals from the start with three trees, and which produces more general $3T2$-graphs, could be more efficient. (The calculation of nests is more rapid if none of the independent sets is too big.)

(2) Characterize those $6T3$-graphs which are generically 3-isostatic.

## 12. Bibliography

[A] Leonard Asimov and Ben Roth, *Rigidity of Graphs*, I: **Trans. Amer. Math. Soc. 245** (1978), 279-289, and II: **J. Math. Anal. Appl. 68** (1979), 171-190.

[E] Jack Edmonds, *Minimum partition of a matroid into independent subsets*, **J. of Res. of the Nat. Bur. of Standards, 69B** (1965), 67-72.

[G] Etienne Guyon, Stéphane Roux, Alex Hansen, Henry Crapo, Daniel Bideau, Jean-Paul Troadec, *Non-local and non-linear problems in mechanics of disordered systems: application to granular media and rigidity problems*, **Reports on Progress in Physics 53**, 1990, 373-419.

[G&W] Harold N. Gabow and Herbert H. Westermann, *Forests, Frames and Games: Algorithms for Matroid Sums and Applications*, **Proceedings of the 20th Annual ACM Symposium on the Theory of Computing**, ACM-0-89791-264-0/0/88/0005/0407, Association of Computing Machinery, 1988.

[I] Hiroshi Imai, *On Combinatorial structures of line drawings of polyhedra*, **Discrete Applied Mathematics 10** (1985), 79-92.

[$L_1$] Gerard Laman, On graphs and rigidity of plane skeletal structures, **J. of Engineering Mathematics 4** (1970), 331-340.

[$L_2$] Laszlo Lovasz and Yechiam Yemeni, *On Generic rigidity in the plane*, **SIAM J. Alg. Disc. Meth. 3** (1982), 91-98.

[$R_1$] Andras Recski, *A Network approach to the rigidity of skeletal structures* II, **Discrete Applied Math. 8** (1984), 63-68.

[$R_2$] Stéphane Roux, Alex Hansen and Etienne Guyon, *Propogation of order in the dilute antiferromagnetic 3-state Potts model*, preprint, Laboratoire d'Hydrodynamique et de Mécanique Physique at the Ecole Supérieure de Physique et Chimie Industrielles de la Ville de Paris, 10 rue Vauquelin, 75231 Paris Cédex 05 France, 1988.

[S] Kokichi Sugihara, *A Unifying Approach to descriptive geometry and mechanisms*, **Disc. Appl. Math. 5** (1983), 313-328.

[$T_1$] Tiong-Seng Tay and Walter Whiteley, *Generating Isostatic Frameworks*, **Structural Topology 11** (1985), 21-69.

[$T_2$] William T. Tutte, *On the problem of decomposing a graph into n connected factors*, **J. London Math. Soc. 36** (1961), 221-230.



Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249 - 6399